

Remote Agent Demonstration

Gregory A. Dorais

Caelum Research,
NASA Ames Research Center
MS 269-2, Moffett Field, CA 94035
650-604-4851

gadorais@ptolemy.arc.nasa.gov

James Kurien

NASA Ames Research Center
MS 269-2, Moffett Field, CA 94035
650-604-4745

kurien@ptolemy.arc.nasa.gov

Kanna Rajan

Caelum Research,
NASA Ames Research Center
MS 269-2, Moffett Field, CA 94035
650-604-0573

kanna@ptolemy.arc.nasa.gov

ABSTRACT

We describe the computer demonstration of the Remote Agent Experiment (RAX). The Remote Agent is a high-level, model-based, autonomous control agent being validated on the NASA Deep Space 1 spacecraft.

Keywords

Model-based autonomous agents, model-based inference, executives, planners, spacecraft.

1. INTRODUCTION

The Remote Agent (RA) is autonomous control software that uses models to reason about the system that it controls and the environment it is in. It does so to accomplish goals over extended periods including diagnosing and recovering from failures without contact with human operators. RA is being validated on the NASA Deep Space 1 spacecraft (DS1) during the Remote Agent Experiment (RAX) scheduled for mid-May, 1999. During RAX, RA will control DS1 and perform several activities including taking pictures, thrusting the ion propulsion engine, and diagnosing and recovering from simulated failures. RA, its major components, and RAX have been described in several papers [1][5][6][7][8][9]. This paper describes a computer demonstration that was designed to aid people unfamiliar with spacecraft and autonomous agent technologies to better understand RA and RAX.

2. REMOTE AGENT ARCHITECTURE

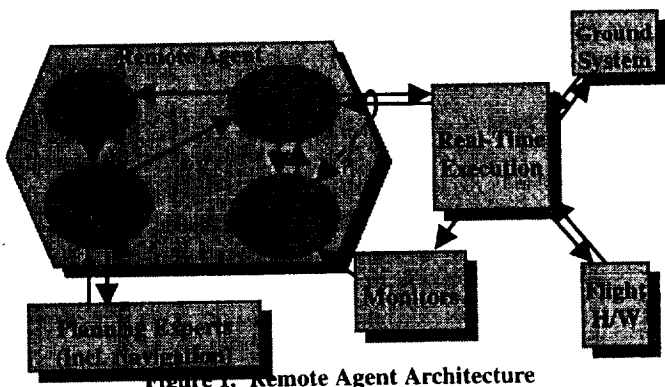


Figure 1. Remote Agent Architecture

As illustrated in figure 1, RA consists of four components: the

Planner/Scheduler (PS), the Mission Manager (MM), the Smart Executive (Exec), and the Mode Identification and Reconfiguration module (MIR).

2.1 Planner/Scheduler and Mission Manager

The Planner/Scheduler (PS) generates the plans that RA uses to control the spacecraft [5]. Given the initial spacecraft state and goals, PS generates a set of synchronized high-level activities that, once executed, will achieve the goals. Mission goals are maintained by MM [1].

PS consists of a heuristic chronological-backtracking search operating over a constraint-based temporal database [5]. PS begins with an incomplete plan and expands it into a complete plan by posting additional constraints in the database. These constraints originate from the goals and from constraint templates stored in a model of the domain. PS consults domain-specific planning experts to access information that is not in its model. The temporal database and the facilities for defining and accessing model information during search are provided by the HSTS system [4].

2.2 Smart Executive

Exec is a reactive, goal-achieving, control system that is responsible for:

- Requesting and executing plans from the planner
- Requesting and executing failure recoveries from MIR
- Executing goals and commands from human operators
- Managing system resources
- Configuring system devices
- Reach and maintain an appropriate safe-mode as necessary
- System-level fault protection

Exec is goal-oriented rather than command-oriented. We define a goal as a state of the system being controlled that must be maintained for a specified length of time. For example, consider the goal: keep device A on from time x to time y. If Exec were to detect that device A is off during that period, it would perform all the commands necessary to turn it back on. This ability is particularly useful in hostile environments where exogenous events can cause devices to behave unpredictably.

Exec controls multiple processes in order to coordinate the simultaneous execution of multiple goals that are often inter-dependent. In order to execute each goal, Exec uses a model-based approach to create a command procedure, which is often complex, designed to robustly achieve the goal.

2.3 Mode Identification/Reconfiguration

The Livingstone inference engine provides the mode identification (MI) and mode reconfiguration (MR) functionality in MIR. To track the modes of system devices, Livingstone eavesdrops on commands that are sent to the spacecraft hardware by the Exec. As each command is executed, Livingstone receives observations from spacecraft's sensors, abstracted by monitors in the spacecraft's control software. Livingstone combines these commands and observations with declarative models of the spacecraft components to determine the current state of the system and report it to the Exec. If any such failures occur, Livingstone will be used to find a repair or workaround that allows the plan to continue execution.

Livingstone uses algorithms adapted from model-based diagnosis [2] to provide the above functions. The key idea underlying model-based diagnosis is that a combination of component modes is a possible description of the current state of the spacecraft only if the set of models associated with these modes is consistent with the observed sensor values. This method does not require that all aspects of the spacecraft state are directly observable, providing an elegant solution to the problem of limited observability.

3. REMOTE AGENT EXPERIMENT

RAX was designed to demonstrate the capabilities of RA on DS1. During RAX, RA will plan how to thrust DS1's ion engine, when to take pictures of asteroids, and when to communicate with Earth. False data will be injected at certain times, unknown to RA, that simulate spacecraft failures. RA will diagnose the cause of these failures and often will be able to find an action that repairs the failure. Otherwise, RA will put the spacecraft into a safe state and find a new plan that accommodates the problem. In addition to operating on its own, RA will demonstrate cooperation with mission controllers by accepting new mission goals and advice on health of the spacecraft.

4. REMOTE AGENT VISUALIZATION

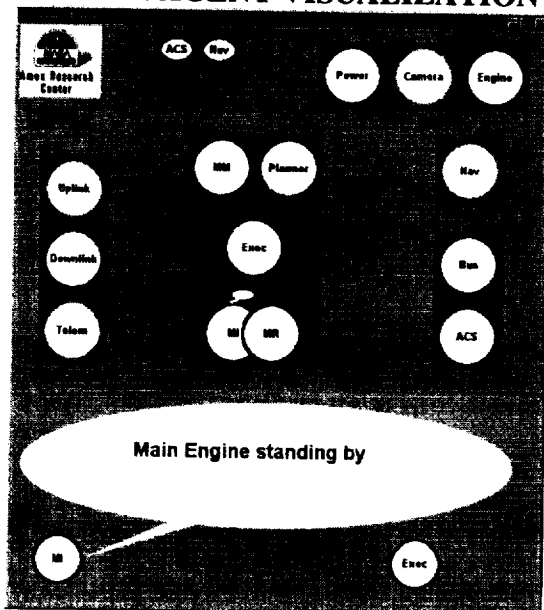


Figure 3. The Remote Agent Demonstration Window

To demonstrate RA, we use a window, in figure 3, that shows the messages as they pass between RA and the other spacecraft software and between RA components. This visualization of the RA can run in real-time while RA is running to show RA's current state, or from a log file of a prior RA run.

The top part of the window has a circle for each component of the RA and spacecraft flight software components RA communicates with. For example, RA sends messages to the attitude control system (ACS) to point the spacecraft toward Earth for communication or toward an asteroid for imaging. A small "speech balloon" travels back and forth between the software components showing which two are currently communicating. In the bottom portion of the window, the current message being transmitted is converted into a simplified English representation. Sensor observations from the spacecraft to RA are shown as moving yellow spheres. In figure 3, MIR is confirming to Exec that the main engine is ready. The demonstration shows a typical 6-day scenario including the ground uplink the command for RA to start its mission, PS interacting with the planning expert modules to create three plans, Exec executing the plans, and MIR sending diagnoses and recoveries to Exec.

5. ACKNOWLEDGMENTS

Our thanks to Bob Kanefsky for developing the visualization software for the RAX demo and to the RAX team cited in [1].

6. REFERENCES

- [1] Bernard, D.E., Dorais, G.A., Fry, C., Gamble Jr., E.B., Kanefsky, B., Kurien, J., Millar, W., Muscettola, N., Nayak, P.P., Pell, B., Rajan, K., Rouquette, N., Smith, B., and Williams, B.C. Design of the Remote Agent experiment for spacecraft autonomy. *Procs. of the IEEE Aerospace Conf., Snowmass, CO, 1998.*
- [2] de Kleer, J., and Williams, B. C. Diagnosis With Behavioral Modes. *Procs. of IJCAI-89, 1989.*
- [3] Gat, E., and Pell, B. Abstract Resource Management in an Unconstrained Plan Execution System, *Procs. of the IEEE Aerospace Conf., Snowmass, CO, 1998.*
- [4] Muscettola, N. HSTS: Integrating planning and scheduling, in Fox, M., and Zweben, M., (eds.), *Intelligent Scheduling*, Morgan Kaufman, 1995.
- [5] Muscettola, N., Smith, B., Chien, S., Fry, C., Rabideau, G., Rajan, K., and Yan, D. On-board Planning for Autonomous Spacecraft, *Procs. of i-SAIRAS, July 1997.*
- [6] Muscettola, N., Nayak, P.P., Pell, B., Williams, B.C., Remote Agent: to boldly go where no AI system has gone before. *Artificial Intelligence*, 103(1/2), August, 1998.
- [7] Pell, B., Gamble, E., Gat, E., Keesing, R., Kurien, J., Millar, W., Nayak, P.P., Plaunt, C., and Williams, B.C. A hybrid procedural/deductive executive for autonomous spacecraft. *Procs. of Autonomous Agents, 1998.*
- [8] Pell, B., Gat, E., Keesing, R., Muscettola, N., and Smith, B. Robust periodic planning and execution for autonomous spacecraft. *Procs. of IJCAI-97, 1997.*
- [9] Williams, B. C., and Nayak, P. A model-based approach to reactive self-Configuring systems, *Procs. of AAAI-96, 1996.*